

# PCS2309 – APP DEVELOPMENT

---

**Experiment No. : 7**

**Date: / /2025**

**Title:** Write programs using Java to create Android application having Databases

- For a simple library application.
- For displaying books available, books lend, book reservation. Assume that student information is available in a database

## **Objective:**

To manage a simple library system with books and students.

To store and retrieve data using SQLite (local database).

To display:

Books available

Books lent

Book reservations

To allow issue and reservation of books to students.

To integrate databases with applications

## **Software/Hardware Requirements:**

- **Java Development Kit (JDK)** - JDK 8 or higher (JDK 11 recommended)
  - **Android Studio** - Latest stable version (e.g., Android Studio Hedgehog or above)
  - **Android SDK** - API Level 21 (Lollipop) or higher (Recommended: API 30–34)
  - **Emulator** - Android Virtual Device (AVD) installed via Android Studio
  - **Gradle** - Comes bundled with Android Studio
  - **Database** - SQLite (built-in in Android SDK)
  - **UI Language** - XML (for activity layouts)
-

## Theory / Concept Overview:

### 1. **Android + Java-Based Development:**

- The application is built using **Java** in **Android Studio**, where each screen (Add Book, Issue Book, etc.) is an **Activity**.
- User interfaces are created using **XML layouts**, and activities handle user interactions.

### 2. **Local Storage using SQLite:**

- The app uses **SQLite**, a lightweight, embedded database, to store data such as books, students, and transactions.
- A helper class (SQLiteOpenHelper) manages the creation, upgrade, and access of the database.

### 3. **Core Library Functions:**

- The app supports **adding books, displaying available/lent books, issuing books, and reserving books**.
- Data is managed using standard CRUD operations: **Create, Read, Update, Delete**.

### 4. **Data Flow and Intents:**

- User input is captured through EditText fields and stored in the database.
- Android **Intents** are used to navigate between different activities (screens).

### 5. **User Feedback and Testing:**

- The app provides feedback using **Toast messages** (e.g., “Book issued successfully”).
- It can be tested on either a **physical Android device** or an **emulator** provided by Android Studio.

---

## Problem Statement:

Managing library records manually is time-consuming, error-prone, and inefficient. Traditional methods often lead to issues like misplaced books, difficulty in tracking issued books, and lack of real-time updates on book availability. There is a need for a user-friendly mobile application that simplifies the process of managing books, issuing and returning them, and handling reservations.

This project aims to develop an **Android-based Library Management System** using **Java and SQLite** to digitize and automate basic library functions such as:

- Adding and displaying books,
- Issuing books to students,
- Tracking book availability and lending status,
- Reserving books for future issue.

The application should provide an intuitive interface for users to maintain accurate and up-to-date records in a local SQLite database for offline accessibility

---

## **Algorithm / Logic:**

### **Approach to Build Android Library Management System (Java + SQLite)**

---

#### **1. Create a New Android Project in Android Studio**

- Open Android Studio → New Project → Select Empty Activity.
  - Name it: **LibraryApp**
  - Choose Java as the language and select minimum SDK (API 21 or above).
  - Finish to create your project structure.
- 

#### **2. Design UI Layouts (XML) for All Activities**

- Create individual layout files in res/layout/
    - activity\_main.xml (Home/Dashboard)
    - activity\_add\_book.xml
    - activity\_issue\_book.xml
    - activity\_view\_books.xml
    - activity\_reserve\_book.xml
  - Use UI components: EditText, Button, ListView or RecyclerView, TextView, etc.
- 

#### **3. Create Java Activity Classes**

- For each screen, create a corresponding Java file:
  - MainActivity.java
  - AddBookActivity.java
  - IssueBookActivity.java
  - ViewBooksActivity.java
  - ReserveBookActivity.java

Each activity handles its layout and user interaction logic.

---

#### **4. Create a SQLite Database Helper Class**

- Create a class: LibraryDBHelper.java that extends SQLiteOpenHelper.
  - Inside, define onCreate() to create tables:
    - books, students, issue, reservations (if needed).
  - Use SQLiteDatabase methods to insert, update, query, and delete records.
- 

#### **5. Set Up Navigation Between Activities**

- Use Intent in Java to move between screens:
- Intent intent = new Intent(MainActivity.this, AddBookActivity.class);
- startActivity(intent);

- Attach navigation buttons to actions in MainActivity.
- 

## **6. Implement "Add Book" Functionality**

- Collect Book ID, Title, Year via EditText.
  - On button click:
    - Validate input.
    - Insert data into books table using SQLiteDatabase.insert().
- 

## **7. Implement "Issue Book" Functionality**

- Collect Book ID, Student ID, Issue Date, Due Date.
  - Check if book is available (status = "Available").
  - If so:
    - Insert into transactions table.
    - Update book status to "Lent" in books table.
- 

## **8. Implement "View Books" Functionality**

- Use Cursor to read all records from books table.
  - Display in ListView or RecyclerView.
  - Optionally categorize: Available, Lent, Reserved.
- 

## **9. Implement "Reserve Book" Logic**

- Check if the selected book is already lent.
  - If yes:
    - Allow reservation (update status = "Reserved" or insert into reservations table).
    - Inform user with a message.
- 

## **10. Test the App on Emulator or Physical Device**

- Run the app in the emulator or on your Android phone with USB debugging.
  - Test all flows:
    - Add Book → Issue Book → View Book → Reserve Book.
  - Fix any crashes or bugs.
-

## Sample Input/Output:

9:13



PRATHYUSHA ENGINEERING COLLEGE

admin

...

Login

9:14

### HOME

- Add Book
- Student Registration
- Issue Book
- Reserve Book
- Return Book
- View Books

9:47

### Add Book

Book1

Apache Cordova in Action

2015

Add Book

≡

9:49

Book ID: book 2  
Name: React Native Cook Book  
Year: 2018

5G

9:49

### Register Student

Student id

Student Name

Student Course

Student Batch

Register Student

### Issue Book

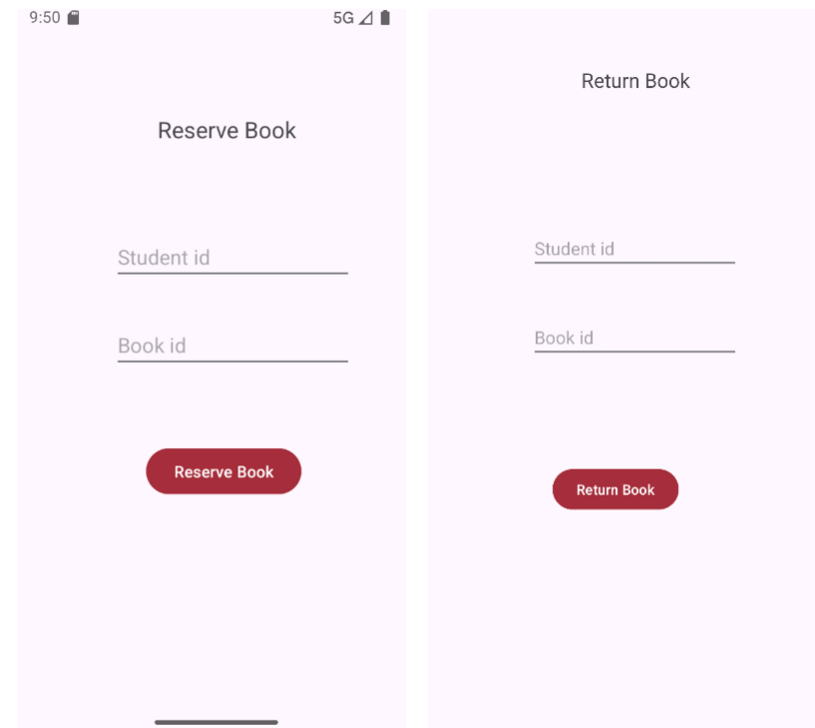
Book Id

Student id

Issue Date

Due Date

Issue



---

## Result:

The application for Library Management System is created using Java.

## Viva Questions:

1. How do you connect your Java code to the XML layout in Android Studio?
2. What is the purpose of the SQLiteOpenHelper class, and how do you use it to manage database operations?
3. How do you navigate between activities and pass data from one screen to another in Android?
4. What steps do you follow to test your app on Android Studio?

Grade / Marks: \_\_\_\_\_

Signature: \_\_\_\_\_